

УДК 519.8

ГРАФОВЫЙ АЛГОРИТМ РЕШЕНИЯ ТРАНСПОРТНЫХ ЗАДАЧ

Ж.А. ЧЕРНЯК, А.А. ЧЕРНЯК

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, 220013, Минск, Беларусь**Международный институт трудовых и социальных отношений
ул. Казинца, 21, к.3, 220099, Минск, Беларусь**Поступила в редакцию 20 марта 2005*

Описан графовый алгоритм решения транспортных задач по критерию стоимости. Благодаря графовой природе алгоритма преодолено традиционное разделение транспортной задачи на вырожденный и невырожденный случаи, а также значительно упрощены математические доказательства сопутствующих результатов. Предложенный подход может быть также использован в учебных программах по математическому программированию.

Ключевые слова: транспортная задача, оргграф, дерево, вырожденность.

Введение

Транспортная задача — задача о наиболее экономном плане перевозок однородного или взаимозаменяемого продукта из пунктов производства в пункты потребления — является важнейшей частной задачей линейного программирования, имеющей обширные практические приложения не только к проблемам транспорта. Транспортная задача выделяется в линейном программировании определенностью экономической характеристики, особенностями математической модели, наличием специфических методов решения.

В данной статье описан графовый алгоритм решения транспортных задач по критерию стоимости. Благодаря графовой природе алгоритма удалось: а) преодолеть традиционное разделение транспортной задачи на вырожденный и невырожденный случаи; б) значительно упростить математические доказательства сопутствующих результатов. Предложенный в статье подход может быть также использован в учебных программах по математическому программированию (именно сложность обоснования вырожденного случая в известных алгоритмах решения транспортных задач служила главным препятствием для включения их в учебные пособия [1–6]).

Предварительные сведения

Предположим, что матрица A системы линейных уравнений $AX^T = C$ имеет ранг r , а множество решений этой системы не пусто. Тогда существует равносильная система вида

Рассмотрим систему уравнений (1). Для упрощения индексации переобозначим свободные переменные x_{j_1}, \dots, x_{j_k} — через z_1, \dots, z_s , а базисные переменные x_{i_1}, \dots, x_{i_r} — через y_1, \dots, y_r . Опишем процедуру жордановой перестановки пары переменных y_l и y_s , предварительно записав систему (1) в новых обозначениях и в виде таблицы:

	1	- z_1	...	- z_s	...	- z_s
$y_1 =$	b_{10}	b_{11}	...	b_{1s}	...	b_{1k}
...
$y_r =$	b_{r0}	b_{r1}	...	b_{rs}	...	b_{rk}

(5)

От этой табличной записи можно легко перейти к обычной — для этого надо скалярно умножить верхнюю строку свободных переменных на каждую из строк таблицы, приравняв к соответствующим переменным y_1, \dots, y_r . Преобразование таблицы (5) в другую таблицу, при котором y_l выводится из базиса переменных, а z_s переходит в него, называется жордановой перестановкой переменных y_l и z_s или жордановой перестановкой с разрешающим элементом b_{ls} . При этом строка и столбец, содержащие разрешающий элемент, также называются разрешающими.

Лемма 1 [6]. Если $b_{ls} \neq 0$, то результатом жордановой перестановки переменных y_l и z_s , примененной к таблице (5), является таблица, которая получается из (5) следующим образом: переменные y_l и z_s меняются местами; элементы разрешающей строки, кроме разрешающего элемента, делятся на него; элементы разрешающего столбца, кроме разрешающего элемента, делятся на него и меняют знак на противоположный; прочие элементы b'_{ij} , где

$$i \neq l, j \neq s, \text{ рассчитываются по формуле } b'_{ij} = b_{ij} - \frac{b_{lj}b_{is}}{b_{ls}}.$$

Рассмотрим теперь задачу ЛП (3) и предположим, что каким-то образом уже получена система (1), равносильная системе уравнений $AX^T = C$, при этом целевая функция f имеет вид $f = f_0 - (f_1x_{j_1} + \dots + f_kx_{j_k})$ и все базисные координаты b_{10}, \dots, b_{r0} неотрицательны, т.е. решение $x_{i_1} = b_{10}, \dots, x_{i_r} = b_{r0}, x_{j_1} = \dots = x_{j_k} = 0$ является базисным планом α задачи (2.3). В этом случае задача (3) эквивалентна таблице

	1	- x_{j_1}	...	- x_{j_k}
$x_{i_1} =$	b_{10}	b_{11}	...	b_{1k}
...
$x_{i_r} =$	b_{r0}	b_{r1}	...	b_{rk}
$f =$	f_0	f_1	...	f_k

(6)

(если при этом не забывать, что все переменные должны принимать только неотрицательные значения). Таблицу (6) назовем s -таблицей с соответствующим ей базисным планом α . Ее первый столбец, содержащий базисные координаты, будем называть p -столбцом, а ее последнюю строку — f -строкой.

Если предположить, что $r = t$, то, принимая во внимание матричные выражения (2) и (4), s -таблицу (6) можно представить в компактном таблично-матричном виде:

	1	$-X_N$	
X_B	$A_B^{-1}C$	$A_B^{-1}A_N$	
$f =$	ПС	$PA_N - D_N$	(7)

Лемма 2 [6]. 1. Если в f -строке s -таблицы (6) все элементы, не считая f_0 , неотрицательны, то α — оптимальный план задачи (3).

2. Если некоторый столбец в s -таблице (6), отличный от p -столбца, содержит в f -строке отрицательный элемент, а все остальные его элементы не положительны, то целевая функция задачи (3) неограниченно возрастает на множестве планов.

3. Если некоторый столбец в s -таблице (6), отличный от f -столбца, содержит в f -строке отрицательный элемент, а среди остальных его элементов имеется положительный, то с помощью жордановой перестановки, примененной к (2.6), можно перейти к базисному плану α' , отличному от α такому, что $f(\alpha') \geq f(\alpha)$.

Из этой леммы можно заключить, что при наличии s -таблицы, которой соответствует некоторый базисный план α^0 задачи (3), можно либо определить оптимальность этого плана, либо установить неограниченность целевой функции на множестве планов, либо жордановой перестановкой перейти к плану α^1 такому, что $f(\alpha^1) \geq f(\alpha^0)$. Таким образом, можно построить последовательность $\alpha^0, \alpha^1, \dots, \alpha^i, \dots$ базисных планов задачи (3), для которой $f(\alpha^0) \leq f(\alpha^1) \leq \dots \leq f(\alpha^i) \leq \dots$

Следующий алгоритм формализует процедуру построения такой последовательности.

Алгоритм *SimpAnticycle*.

Цикл 0. Задать начальную s -таблицу, которой соответствует некоторый базисный план α^0 задачи (3). Объявить эту s -таблицу активной и перейти к следующему циклу.

Цикл k ($k \geq 1$). Рассмотреть элементы f -строки активной s -таблицы. Если все они, не считая первого, неотрицательны, то соответствующий базисный план оптимален, и алгоритм завершает свою работу.

Если в f -строке, не считая ее первого элемента, есть отрицательный элемент, то рассмотреть столбец, содержащий этот элемент. В случае наличия нескольких таких отрицательных элементов следует выбрать тот из них, который соответствует переменной x_s с минимально возможным индексом s (правило «антизацикливания» Блэнда при выборе разрешающего столбца). Пусть $b_{1s}, b_{2s}, \dots, b_{rs}, f_s$ — элементы выбранного столбца. Если все они не положительны, то целевая функция задачи (3) неограниченно возрастает на множестве планов и алгоритм завершает свою работу.

Если в рассматриваемом столбце найдется положительный элемент, то произвести жорданову перестановку. В качестве разрешающего выбрать элемент b_{is} , для которого выполняется

$$\text{равенство: } \frac{b_{i0}}{b_{is}} = \min \left\{ \frac{b_{io}}{b_{is}} : \text{при } b_{is} > 0 \right\}, \text{ где } b_{10}, b_{20}, \dots, b_{r0}, f_0 \text{ — элементы } p\text{-столбца активной}$$

s -таблицы. Другими словами, для всех положительных элементов b_{is} разрешающего столбца

составить отношения $\frac{b_{i0}}{b_{is}}$ и выбрать среди них наименьшее отношение $\frac{b_{i0}}{b_{is}}$. В случае неодно-

значности выбора (наличие нескольких строк, для которых достигается минимум

$$\min \left\{ \frac{b_{i0}}{b_{is}} : \text{при } b_{is} > 0 \right\}, \text{ следует выбрать ту из них, которая соответствует переменной } x_l \text{ с ми-}$$

нимально возможным индексом l (правило "антизацикливания" Блэнда при выборе разрешающей строки). Перейти к следующему циклу, объявив активной s -таблицу, полученную в результате жордановой перестановки.

Теорема 1 [7, 8]. Алгоритм *SimpAnticycle* за конечное число циклов либо находит оптимальный план задачи (3), либо устанавливает неограниченность целевой функции на множестве планов.

Известны различные доказательства конечности алгоритмов симплекс-метода, использующих правило антициклирования Блэнда. Здесь мы приведем элементарное обоснование этого результата, не прибегая к дополнительным сведениям или утверждениям.

Доказательство теоремы 1. Вначале докажем, что ни одна s -таблица в ходе алгоритма *SimpAnticycle* не может повториться. Предположим противное: в ходе этого алгоритма порождается такая последовательность s -таблиц T_m, T_{m+1}, \dots, T_d , в которой $T_m = T_d$. Если теперь $\alpha^m, \alpha^{m+1}, \dots, \alpha^d$ — соответствующие им базисные планы, то $\alpha^m = \alpha^d$. По лемме 2 $f(\alpha^m) \leq f(\alpha^{m+1}) \leq \dots \leq f(\alpha^d)$. Поэтому $f(\alpha^m) = f(\alpha^{m+1}) = \dots = f(\alpha^d)$.

Удалим из s -таблиц T_m, T_{m+1}, \dots, T_d все строки и столбцы с переменными, не участвующими в жордановых перестановках при переходе от T_i к T_{i+1} , $i = m, m+1, \dots, d-1$. Получим последовательность M_m, M_{m+1}, \dots, M_d модифицированных таблиц, в которой $M_m = M_d$. Из вида жордановой перестановки переменных x_i и x_s (лемма 1) следует, что элемент на пересечении p -столбца и f -строки остается неизменным, если и только если элемент на пересечении p -столбца и разрешающей строки (с переменной x_i) равен нулю; при этом неизменными остаются также и все другие элементы p -столбца. Поскольку в модифицированных таблицах присутствуют только строки с переменными, участвующими в жордановых перестановках при переходе от T_i к T_{i+1} , $i = m, m+1, \dots, d-1$, то все элементы p -столбцов этих таблиц, не считая элементов f -строки, равны нулю.

Пусть x_q — переменная с наибольшим индексом q среди всех переменных, содержащихся в строках и столбцах M_m, M_{m+1}, \dots, M_d . Пусть M_a — первая из этих таблиц, чья свободная переменная x_q переходит в базис переменных следующей за M_a таблицы M_{a+1} . Пусть M_c — первая после M_{a+1} таблица, которая преобразуется в таблицу M_{c+1} жордановой перестановкой переменных x_q и x_p , причем x_q становится свободной переменной в таблице M_{c+1} . Таблицы M_a и M_c удобно представить в следующем виде:

$$M_a :$$

	1	$-z_1$	\dots	$x_q = -z_j$	\dots	$-z_l$
$y_{i=}$	0	a_{i1}	\dots	a_{ij}	\dots	a_{il}
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$y_{s=}$	0	a_{s1}	\dots	a_{sj}	\dots	a_{sl}
$f=$	f_0	f_1	\dots	f_j	\dots	f_l

$$M_c :$$

	1	$-v_1$	\dots	$x_p = -v_t$	\dots	$-v_l$
$w_1 =$	0	c_{11}	\dots	c_{1t}	\dots	c_{1l}
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_q = w_i =$	0	c_{i1}	\dots	c_{it}	\dots	c_{il}
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$w_s =$	0	c_{s1}	\dots	c_{st}	\dots	c_{sl}
$f=$	f_0	g_1	\dots	g_t	\dots	g_l

из V . Тогда пара (V, E) называется ориентированным графом или орграфом; при этом элементы множества V называются вершинами, а элементы множества E — дугами этого орграфа.

Если пара вершин u и w составляет дугу, то u и w называются концевыми вершинами этой дуги. В дуге (v, w) вершина v называется началом, а вершина w — концом этой дуги. Иногда удобно говорить, что дуга (v, w) выходит из v и входит в w . Концевые вершины дуги называются смежными. Смежными будут также называться дуги, имеющие общие концевые вершины.

Пусть дан граф $G=(V, E)$. Последовательность $P=e_1, e_2, \dots, e_r$ различных дуг из E называются путем длины r в G , если каждая дуга $e_i \in P$, отличная от первой e_1 и последней e_r , имеет общие концевые вершины с соседними дугами e_{i-1} и e_{i+1} , причем одну с e_{i-1} , а другую — с e_{i+1} . Путь P называется простым, если каждая его дуга e_i , отличная от первой и последней, смежна только с двумя дугами этого пути, а именно с соседними e_{i-1} и e_{i+1} .

Обозначим через s концевую вершину первой дуги e_1 пути P , не принадлежащей соседней дуге e_2 , а через t — концевую вершину последней дуги e_r пути P , не принадлежащей соседней дуге e_{r-1} . В этом случае говорят, что путь P соединяет вершины s и t или является (s, t) -путем. Если вершины s и t совпадают, то путь P называют циклом.

Если начало дуги e_i , где $i \neq 1, r$, принадлежит предшествующей дуге e_{i-1} , а конец — последующей дуге e_{i+1} , то дуга e_i называется прямой в P , в противном случае e_i называется обратной в P ; прямыми также называются: e_1 , если s — ее начало, и дуга e_r , если t — ее конец; в противном случае e_1 и e_r называются обратными в P . Итак, все дуги пути P в орграфе G можно разбить на два множества P^+ и P^- прямых и обратных дуг соответственно. Путь P , состоящий только из прямых дуг, называется орпутем. В этом случае удобно называть P орпутем из s в t . Если вершины s и t совпадают, то орпуть называется орциклом. Орграф, не содержащий орциклов, называется бесконтурным.

Пусть дан орграф $G=(V, E)$ с множеством вершин $V=\{v_1, \dots, v_n\}$ и множеством дуг $E=\{e_1, \dots, e_m\}$. Матрицей инцидентностей графа G называется матрица $Ai(G)$ размера $n \times m$, определяемая следующим образом: в этой матрице на позиции (i, k) находится элемент a_{ik} , равный

$$\begin{cases} 1, & \text{если } e_k \text{ выходит из вершины } v_i; \\ -1, & \text{если дуга } e_k \text{ входит в вершину } v_i; \\ 0, & \text{в противном случае.} \end{cases}$$

Очевидно, вектор-строки матрицы $Ai(G)$ линейно зависимы, поскольку их сумма равна нулевому вектору. В ряде случаев нам придется удалять строки из $Ai(G)$ с тем, чтобы результирующая матрица имела ранг, равный числу ее строк (такие матрицы будем называть приведенными).

Одновершинный граф называется тривиальным. Граф называется связным, если он либо тривиальный, либо любые две его вершины могут быть соединены путем. Граф T называется подграфом графа G , если вершины и дуги графа T являются также вершинами и дугами графа G . Если при этом множество вершин подграфа T совпадает с множеством вершин графа G , то T называется остовным подграфом графа G .

Всякий максимальный связный подграф графа G называется его компонентой связности. Слово "максимальный" означает максимальный относительно включения, т.е. не содержащийся в связном подграфе с большим числом вершин или дуг.

Связный граф без циклов называется деревом, несвязный граф без циклов называется лесом. Если остовный подграф T графа G является деревом, то T называется остовным деревом графа G . Вершина в графе называется листом, если она является концевой вершиной только одной дуги.

Лемма 4 [6]. *Любое нетривиальное дерево содержит, по крайней мере, два листа.*

Лемма 5 [6]. *Пусть $T=(V, E)$ — дерево; C — единственный цикл графа $T+e$, где e — некоторая дуга. Тогда при удалении любой дуги цикла C в $T+e$ вновь получится дерево с множеством вершин V .*

Лемма 6 [6]. *Для графа G , имеющего n вершин и m дуг, следующие утверждения эквивалентны:*

1. G — дерево;
2. G — связный граф и $m=n-1$;
3. G не содержит циклов и $m=n-1$;
4. любые две вершины графа G соединены единственным простым путем;
5. G не содержит циклов, и после соединения дугой двух его несмежных вершин вновь полученный граф будет содержать ровно один цикл.

Граф называется двудольным, если множество его вершин можно так разбить на два непересекающихся подмножества, называемых долями, что любые две смежные вершины будут находиться в разных долях.

Лемма 7 [6]. *Граф является двудольным, если и только если он либо является лесом, либо все его циклы состоят из четного числа дуг.*

Основные результаты

Пусть имеется p пунктов u_1, \dots, u_p производства однородной продукции, причем объем производства в пункте u_i равен a_i , $i = 1, \dots, p$. Вся произведенная продукция потребляется в q пунктах w_1, \dots, w_q , причем спрос в пункте w_k составляет b_k , $k = 1, \dots, q$. Предполагается выполнимость балансового соотношения: $a_1 + \dots + a_p = b_1 + \dots + b_q$. Известны стоимости – тарифы – d_{ik} перевозок единицы продукции из пункта u_i в пункт w_k . Транспортная задача по критерию стоимости состоит в нахождении такого плана перевозок из пунктов производства в пункты потребления, который не допускает затоваривания, удовлетворяет весь спрос и при этом минимизирует суммарную стоимость перевозок. Транспортная задача является частным случаем задачи об оптимальном потоке. Однако ее специфические особенности открывают более эффективный и наглядный путь к решению задачи с помощью графовой модификации симплекс-метода.

Запишем теперь транспортную задачу по критерию стоимости, как задачу ЛП:

$$f(x) = -DX^T \rightarrow \max$$

на множестве планов

$$\begin{cases} Ai(G)X^T = C, \\ X \geq 0, \end{cases} \quad (8)$$

где $X = (x_{11}, x_{12}, \dots, x_{1q}, x_{21}, \dots, x_{2q}, \dots, x_{p1}, \dots, x_{pq})$, $D = (d_{11}, d_{12}, \dots, d_{1q}, d_{21}, \dots, d_{2q}, \dots, d_{p1}, \dots, d_{pq})$, $C^T = (a_1, \dots, a_p, -b_1, \dots, -b_q)$, $a_1 + \dots + a_p = b_1 + \dots + b_q$, $m = p + q - 1$, $Ai(G)$ — матрица инцидентностей двудольного орграфа $G=(U \cup W, E)$ с долями $U = \{u_1, \dots, u_p\}$ и $W = \{w_1, \dots, w_q\}$ и с множеством дуг $E = \{(u_i, w_k), i=1, \dots, p, k=1, \dots, q\}$.

Условимся, что столбцы и строки матрицы инцидентностей $Ai(G)$ расположены в соответствии со следующим порядком дуг в E и вершин в $U \cup W$:

$$E = \{(1,1), (1,2), \dots, (1,q), (2,1), \dots, (2,q), \dots, (p,1), \dots, (p,q)\}, U \cup W = \{u_1, \dots, u_p, w_1, \dots, w_q\}$$

(всюду под записью (i,k) подразумевается дуга (u_i, w_k)).

Столбец или строка произвольной матрицы M , соответствующей переменной x_{ik} , будут соответственно называться $\langle i,k \rangle$ -столбцом или $\langle i,k \rangle$ -строкой и обозначаться $M^{\langle i,k \rangle}$ или $M_{\langle i,k \rangle}$ соответственно. Индексом или номером переменной x_{ik} считается пара (i,k) .

При использовании двойной индексации, например, в переменных x_{ik} или координатах α_{ik} , фраза "индекс (i,k) меньше индекса (r,j) " будет означать либо $i < r$, либо $i=r, k < j$.

Лемма 8 [6]. *Ранг матрицы $Ai(G)$ в задаче (8) равен m , причем любые m строк этой матрицы линейно независимы.*

Известно, что множество планов в задаче (8) не пусто. Поэтому ранг расширенной матрицы $(Ai(G), C)$ равен рангу m матрицы $Ai(G)$. Следовательно, одно из уравнений в системе уравнений $Ai(G) \cdot X^T = C$ избыточно, т.е. является следствием других. По лемме 8 любые m

строк матрицы $Ai(G)$ линейно независимы. Поэтому в качестве избыточной можно взять любую строку в $(Ai(G), C)$. Условимся, что это будет последняя строка, соответствующая вершине w_q .

Через \bar{A} и \bar{C} будем обозначать матрицы, полученные из $A=A_i(G)$ и C соответственно удалением последней строки. Таким образом, вместо задачи (8) можно рассматривать эквивалентную с тем же многогранником планов задачу

$$\begin{aligned} f(x) = -DX^T \rightarrow \max \\ \text{на множестве планов} \\ \begin{cases} \bar{A}X^T = \bar{C}, \\ X \geq 0 \end{cases} \end{aligned} \quad (9)$$

(здесь ранг m матрицы \bar{A} равен числу ее строк).

Пусть B — некоторый набор двойных индексов. Через M_B обозначим подматрицу матрицы M , составленную из всех таких столбцов $M^{<i,k>}$, для которых $(i,k) \in B$. Графом T_B , индуцированным набором B , назовем двудольный орграф с долями $U=\{u_1, \dots, u_p\}$ и $W=\{w_1, \dots, w_q\}$, содержащий дугу (i,k) , если и только если $(i,k) \in B$.

Лемма 9. Пусть α — план задачи (9) с не более, чем m положительными координатами, $B(\alpha)$ — некоторый набор из двойных индексов, содержащий номера всех положительных координат плана α . Тогда α — базисный план с множеством номеров базисных координат $B(\alpha)$, если и только если $T_{B(\alpha)}$ — дерево.

Доказательство. Предположим, что $T_{B(\alpha)}$ не является деревом. Тогда по лемме 6 орграф $T_{B(\alpha)}$ содержит цикл, поскольку число дуг в нем на 1 меньше числа вершин. Пусть это будет цикл $(i,i), (i+1,i), (i+1,i+1), \dots, (k,k-1), (k,k), (i,k)$. Из строения столбцов матрицы $A = Ai(G)$ следует, что сумма $A^{<i,i>} - A^{<i+1,i>} + A^{<i+1,i+1>} - \dots + A^{<k,k>} - A^{<i,k>}$ равна нулевому столбцу. А это означает линейную зависимость этих столбцов. Но тогда линейно зависимы все столбцы матрицы $A_{B(\alpha)}$ и, следовательно, столбцы матрицы $\bar{A}_{B(\alpha)}$. Поэтому матрица $\bar{A}_{B(\alpha)}$ необратима и ввиду этого $B(\alpha)$ не может быть множеством номеров базисных координат какого-либо плана задачи (9).

Предположим теперь, что $T_{B(\alpha)}$ — дерево, но столбцы матрицы $\bar{A}_{B(\alpha)}$ линейно зависимы. Тогда найдутся ненулевые числа $\lambda_{i_1 k_1}, \dots, \lambda_{i_r k_r}$ такие, что

$$0 = \lambda_{i_1 k_1} \bar{A}^{<i_1, k_1>} + \dots + \lambda_{i_r k_r} \bar{A}^{<i_r, k_r>}, \quad (10)$$

где $B' = \{(i_1, k_1), \dots, (i_r, k_r)\} \subseteq B(\alpha)$, $r \leq m$. Так как $T_{B'}$ — подграф дерева $T_{B(\alpha)}$, то $T_{B'}$ — лес. Поэтому в силу леммы 4 в $T_{B'}$ найдется лист, отличный от вершины w_q . Пусть это будет вершина u_{i_1} . Тогда в $\bar{A}_{B'}$ только $<i_1, k_1>$ -столбец содержит ненулевой элемент в i_1 -й строке. Следовательно, (10) может выполняться только при $\lambda_{i_1 k_1} = 0$. Противоречие. Таким образом, столбцы в матрице $\bar{A}_{B(\alpha)}$ линейно независимы. Отсюда $B(\alpha)$ — множество номеров базисных координат базисного плана α . Лемма доказана.

На лемме 9 основано несколько известных алгоритмов построения базисного плана задачи (9). Приведем один из них.

Алгоритм *MinTarif*.

Шаг 0. Положить $\alpha = (\alpha_{11}, \alpha_{12}, \dots, \alpha_{pq}) = (0, 0, \dots, 0) \in Af^{pq}$. Задать двудольный граф H без дуг с долями $U = \{u_1, \dots, u_p\}$, $W = \{w_1, \dots, w_q\}$. Вершинам u_i и w_k приписать числа a_i и b_k соответственно, обозначив их через $d(u_i)$ и $d(w_k)$ и назвав $d(u_i)$ и $d(w_k)$ весами вершин u_i и w_k , $i = 1, \dots, p$, $k = 1, \dots, q$. Перейти к следующему шагу.

Шаг r ($r \geq 1$).

1. Если не все вершины имеют нулевые веса, то среди всех таких вершин выбрать пару (u_i, w_k) , которой соответствует минимально возможный тариф d_{ik} . Добавить дугу (i, k) , положив ее вес равным числу $s = \min\{d(u_i), d(w_k)\}$. Изменить (i, k) -ю нулевую координату α_{ik} точки α , положив ее равной s . Изменить также веса вершин u_i и w_k , положив их соответственно равными $d(u_i) - s$ и $d(w_k) - s$. Перейти к следующему шагу.

2. Если все вершины имеют нулевые веса и $r < m + 1$, то добавить к H ту дугу (i, k) , для которой вершины u_i и w_k принадлежат разным компонентам связности. Присвоить дуге (i, k) вес 0 и перейти к следующему шагу.

3. Если все вершины имеют нулевые веса и $r = m + 1$, то алгоритм завершает работу и точка α является искомым базисным планом задачи (8).

Теорема 2. Точка α , построенная алгоритмом *MinTarif* к моменту завершения его работы, является базисным планом задачи (8).

Доказательство. Первоначально суммы весов вершин долей U и W равны между собой, поскольку $\sum_{i=1}^p a_i = \sum_{k=1}^q b_k$. Затем на каждом шаге эти суммы уменьшаются на одинаковую величину, причем по меньшей мере на одну вершину с нулевым весом становится больше. По завершению алгоритма сумма весов вершин хотя бы одной из долей должна стать нулевой. Следовательно, таковой будет и сумма весов вершин другой доли. Это означает выполнимость ограничений по предложению и спросу для построенной алгоритмом точки α . Таким образом, α — план задачи (9).

Докажем, что в процессе работы алгоритма не могут появляться циклы. Предположим, что еще до того момента, когда веса всех вершин станут нулевыми, на каком-то шаге алгоритма граф H содержит простой цикл (i, i) , $(i + 1, i)$, $(i + 1, i + 1)$, ..., (k, k) , (i, k) . Так как в цикле все дуги равноправны, то не теряя общности можно считать, что именно дуга (i, i) появилась в H раньше других дуг этого цикла. Сразу после появления дуги (i, i) вес по крайней мере одной из вершин u_i , w_i станет нулевым. А это означает, что на последующих шагах ни одна из добавляемых дуг цикла уже не может иметь концевой эту вершину, т. е. появление хотя бы одной из дуг $(i + 1, i)$, (i, k) невозможно.

Противоречие.

Не могут появиться циклы и после обнуления весов всех вершин, поскольку на каждом последующем шаге (если таковые будут) добавляется дуга с концевыми вершинами из разных компонент связности.

Таким образом, по завершению алгоритма граф H не содержит циклов и имеет m дуг и $m + 1$ вершин, т.е. по лемме 6 является деревом. Если обозначить теперь через $B(\alpha)$ множество номеров координат α_{ik} плана α , соответствующих дугам (i, k) построенного алгоритмом дерева H , то $H = T_{B(\alpha)}$. Отсюда в силу леммы 9 α — базисный план. Теорема доказана.

Предположим, что построен начальный базисный план α задачи (9) с множеством B номеров базисных координат. Запишем в матричном виде (см. (9)) начальную s -таблицу, которой соответствует план α :

$$\begin{array}{|c|c|c|} \hline & 1 & -X_N \\ \hline X_B = & \overline{\overline{A_B C}} & \overline{\overline{A_B A_N}} \\ \hline f = & -\overline{\overline{P C}} & D_N - \overline{\overline{P A_N}} \\ \hline \end{array}, \quad (11)$$

где N — множество номеров свободных переменных, $\alpha_B = \overline{\overline{A_B^{-1} \cdot C}}$, $\overline{\overline{P}} = D_B \cdot \overline{\overline{A_B^{-1}}}$.

Рассмотрим теперь двудольный орграф T_B , который в силу леммы 9 является деревом.

Согласно лемме 6, любые две вершины в u_i, w_k в дереве T_B можно соединить единственным простым (u_i, w_k) -путем, который обозначим P_{ik} . Зададим некоторый порядок дуг в дереве T_B и определим матрицу P размера $m \times (pq)$, строки которой соответствуют дугам дерева T_B , а столбцы — путям P_{ik} : элемент на пересечении $\langle l, r \rangle$ -строки и $\langle i, k \rangle$ -столбца равен 1, если дуга (l, r) — прямая в P_{ik} , равен — 1, если дуга (l, r) — обратная в P_{ik} , и равен 0, если P_{ik} не содержит дуги (l, r) .

Возьмем, к примеру, произвольную свободную переменную x_{ik} , т. е. $(i, k) \in N$ и, следовательно, (i, k) не является дугой в T_B . Не теряя общности, можно считать, что

$$P_{ik} = (i, i), (i+1, i), (i+1, i+1), \dots, (k, k-1), (k, k)$$

Тогда $\langle i, k \rangle$ -столбец $P_N^{(i, k)}$ в матрице P_N (при соответствующей перестановке строк) имеет вид

$$P_N^{(i, k)} = \begin{pmatrix} b_{ii} \\ b_{i+1, i} \\ b_{i+1, i+1} \\ \vdots \\ b_{k, k-1} \\ b_{k, k} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ \vdots \\ -1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (12)$$

Лемма 10.

$$P_N = \overline{\overline{A_B^{-1} A_N}}.$$

Доказательство. Матрица инцидентностей $Ai(P_{ik})$ пути P_{ik} имеет следующий вид (нулевые строки здесь опускаются):

$$(i, i)(i+1, i)(i+1, i+1) \dots (k, k-1)(k, k)$$

Согласно алгоритму *SimpAnticycle*, необходимо выбрать наименьшее из отношений:

$$\frac{\alpha_{ii}}{b_{ii}} = \alpha_{ii}, \frac{\alpha_{i+1,i+1}}{b_{i+1,i+1}} = \alpha_{i+1,i+1}, \dots, \frac{\alpha_{kk}}{b_{kk}} = \alpha_{kk}. \text{ Пусть, к примеру, наименьшим будет } \alpha_{ii}. \text{ В этом случае}$$

в качестве разрешающего берется элемент $b_{ii} = 1$ и производится жорданова перестановка переменных x_{ii} и x_{ik} . В результате получится новая s -таблица, которой соответствует новый базисный план α' . Фрагмент этой s -таблицы будет выглядеть так:

	1	$-x_{ik}$
$x_{ik} =$	α_{ii}	1
$x_{i+1,i} =$	$\alpha_{i+1,i} + \alpha_{ii}$	1
$x_{i+1,i+1} =$	$\alpha_{i+1,i+1} - \alpha_{ii}$	-1
\vdots	\vdots	\vdots
$x_{k,k-1} =$	$\alpha_{k,k-1} + \alpha_{ii}$	-1
$x_{kk} =$	$\alpha_{kk} - \alpha_{ii}$	1
		0
		\vdots
\vdots	\vdots	0
$f =$		

Так как новое множество номеров B' базисных переменных совпадает с множеством $(B \cup (i, k)) \setminus \{(i, i)\}$, то $T_{B'} = T_B + (i, k) - (i, i)$. По лемме 6 граф $T_B + (i, k)$ содержит единственный цикл $(i, i), (i+1, i), (i+1, i+1), \dots, (k, k-1), (k, k), (i, k)$. Поэтому по лемме 5 $T_{B'}$ — дерево.

Суммируя сказанное, получаем следующий графовый аналог перехода от базисного плана α к новому базисному плану α' (выбор свободной переменной x_{ik} будет описан ниже): пусть каждой дуге (r, j) дерева T_B приписан вес, равный координате α_{rj} плана α ; среди всех прямых дуг (u_i, w_k) -пути P_{ik} выбрать дугу (r, j) наименьшего веса α_{rj} ; изменить вес дуг (u_i, w_k) -пути, прибавив α_{rj} к весам обратных дуг и отняв α_{rj} от весов прямых дуг; удалить дугу (r, j) и добавить дугу (i, k) , положив ее вес равным α_{rj} ; вновь полученное взвешенное дерево $T_{B'}$ будет соответствовать новому базисному плану α' . Описанный переход от дерева T_B к дереву $T_{B'}$ назовем жордановым переходом с добавляемой дугой (i, k) и исключаемой дугой (r, j) .

Теперь остается описать процедуру выбора добавляемой дуги (i, k) , что соответствует выбору свободной переменной x_{ik} в s -таблице (11). Согласно алгоритму *SimpAnticycle*, в s -таблице (11) выбирается столбец, отличный от s -столбца, с отрицательным элементом в матрице-строке $D_N - \bar{P} \bar{A}_N$, где $\bar{P} = P_B \bar{A}_B^{-1}$. Можно избежать обращения матрицы \bar{A}_B для определения \bar{P} , решив вместо этого систему

$$(y_1, \dots, y_p, z_1, \dots, z_q) A_B = D_B \quad (13)$$

m линейных уравнений с $m + 1$ переменными при условии, что $z_q = 0$, поскольку

$(y_1, \dots, y_p, z_1, \dots, z_{q-1}, 0)A_B = D_B \Leftrightarrow (y_1, \dots, y_p, z_1, \dots, z_{q-1})\bar{A}_B = D_B \Leftrightarrow (y_1, \dots, y_p, z_1, \dots, z_{q-1}, 0) = D_B \bar{A}_B^{-1}$
(здесь $A = Ai(G)$). Систему (3.6) можно записать так:

$$\begin{cases} y_r - r_j = d_{rj}, \\ \text{индекс } (r, j) \text{ пробегает все дуги дерева } T_B. \end{cases} \quad (14)$$

Известен простой алгоритм ее решения, называемый методом потенциалов.

Алгоритм *MetPoten*.

Шаг 0. Вершине w_q присвоить потенциал z_q , положив его равным нулю. Перейти к следующему шагу.

Шаг $r (r \geq 1)$. Если $r = m + 1$, то алгоритм завершает работу. Если $r \leq m$, то выбрать дугу (r, j) в T_B такую, что ровно одной из ее концевых вершин присвоен потенциал. Если именно вершине w_j присвоен потенциал z_j , то присвоить вершине u_r потенциал y_r , равный $z_j + d_{rj}$; если именно вершине u_r присвоен потенциал y_r , то присвоить вершине z_j потенциал w_j , равный $y_r - d_{rj}$. Перейти к следующему шагу.

Лемма 11. После завершения алгоритма *MetPoten* всем вершинам дерева T_B будут присвоены потенциалы, которые будут составлять решение системы (14).

Доказательство. Условимся удалять в конце каждого шага ту дугу дерева T_B , которая выбиралась в его начале. Тогда к концу $(r + 1)$ -го шага будет удалено r дуг и образуется лес с $r + 1$ компонентами связности, поскольку удаление дуги из леса увеличивает число компонент связности на 1. Кроме того, все компоненты связности будут содержать ровно по одной вершине, которым уже присвоены потенциалы (это утверждение непосредственно проверяется индуктивно). Таким образом, к началу $(m + 1)$ -го шага будем иметь граф без дуг, состоящий из одновершинных компонент связности с присвоенными этим вершинам потенциалами. По построению эти потенциалы будут составлять решение системы (14).

Итогом приведенных выше рассуждений является графовый алгоритм решения транспортной задачи по критерию стоимости – модификация алгоритма *SimpAnticycle*.

Алгоритм *SimpMainTrans*.

Шаг 0. С помощью алгоритма *MinTarif* построить дерево T_B , веса дуг которого составляют базисные координаты начального базисного плана задачи (8). Перейти к следующему шагу.

Шаг $r (r \geq 1)$.

1. С помощью алгоритма *MetPoten* найти решение Π системы уравнений (9). Вычислить матрицу-строку $D_N - \bar{\Pi} \bar{A}_N$.

2. Если в $D_N - \bar{\Pi} \bar{A}_N$ нет отрицательных элементов, то алгоритм завершает работу и план α — оптимальный. Если в $D_N - \bar{\Pi} \bar{A}_N$ имеются отрицательные элементы, то выбрать среди них элемент α_{ik} с наименьшим индексом (i, k) .

3. Произвести жордановый переход от дерева T_B к дереву $T_{B'}$ с добавляемой дугой (i, k) и исключаемой дугой (r, j) . Если имеется несколько дуг, которые могут быть выбраны в качестве исключаемых, то предпочтение отдать той, у которой индекс наименьший. Веса дуг дерева $T_{B'}$ составят базисные координаты нового базисного плана α' .

4. Переобозначить дерево $T_{B'}$ через T_B , а план α' — через α и перейти к следующему шагу.

GRAPH ALGORITHM FOR SOLVING TRANSPORT PROBLEMS

Z.A. CHERNIAK, A.A. CHERNIAK

Abstract

A graph algorithm for solving transport problems of cost criterion is described. Owing to graph nature of the algorithm the traditional division of transport problems into degenerate and nondegenerate cases are overcome and mathematical proofs are simplified. The offered approach can be used in teaching mathematical programming.

Литература

1. *Костевич Л.С.* Информационные технологии оптимальных решений. Мн., 1999.
2. Исследование операций в экономике / Под ред. проф. Кремера Н.Ш. М., 2000.
3. *Кузнецов А.В., Холод Н.И.* Математическое программирование. Мн., 1984.
4. *Ашманов С.А.* Линейное программирование. М., 1981.
5. *Кузнецов А.В., Сакович В.А., Холод Н.И.* Высшая математика: математическое программирование. Мн., 1994.
6. *Черняк А.А., Новиков В.А., Мельников О.И., Кузнецов А.В.* Математика для экономистов на базе matched. СПб., 2003.
7. *Схрейвер А.* Теория линейного и целочисленного программирования. 1. М., 1991.
8. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность. М., 1985.